Engineering with Computers (2000) 16: 36–52

Engineering with Computers

Abstracting and Exploring Functional Design Information for Conceptual Mechanical Product Design

Y.-M. Deng, S. B. Tor and G. A. Britton

Design Research Centre, School of Mechanical and Production Engineering, Nanyang Technological University, Singapore

Abstract. *The conceptual stage of mechanical product design has been recognised as the most critical product development stage. This is an area which is not addressed by current CAD systems. In this paper, we present our work on the abstraction and exploration of functional design information to support this design area. We consider the exploration of relevant functional design information as an important aspect of design work during this design process. Four aspects of functional design information are abstracted, including function, behaviour, structure and working environment. The working environment information is demonstrated as being useful to the exploration of functional design solutions. The physical behaviour is represented by an input-output flow-of-action, contrary to the commonly-used input-output flow-of-object approach (where object refers to material, energy or signal). Not only is this approach more generic, more importantly, we argue that the input-output flow-of-object is only a physical-level design abstraction, while flow-of-action is a functional-level design abstraction, hence it can capture design intention. Based on this strategy, we explore functional design information by considering the relevant action flows between the design and its working environment, as well as between the different components of the design, including their attributes, attribute expressions, constraints, and so on. Prototype software has been developed to implement the proposed approaches. A simple design example is used to illustrate the various aspects of functional design information and their exploration process.*

Keywords. Design information; Functional design; Input-output flow-of-action; Mechanical product design

1. Introduction

The main focus of the conceptual stage of mechanical product design is to generate a design solution that can fulfil the required functions. Function plays a key role in this process, just as geometry does for the detailed design stage. To conduct functional design [1], it is necessary for the designer to abstract, explore and organise functional design information. We consider this as essential in exploring possible design solutions. This is especially true for developing a computer-based design tool, where a comprehensive design representation model must initially be developed. Furthermore, the functional design information is useful in developing various models, such as a geometric model, kinematic model, dynamic model, etc. [2]. Appropriately incorporating functional design information within these models will provide the meta-knowledge in efficiently applying these models.

Many researchers regard (represent) function as a transformation between the input and output of material, energy or signal [3–10]. For example, Pahl and Beitz [3] characterise function as *a general input/output relation of a system whose purpose is to perform a task.* The input/output relations are defined by flows of energy, material or signals. As has been pointed out by Ullman [11], all methodical design authors use a system transformational definition of function.

We can see that these researches only abstract the input and output as a flow of *objects*, i.e. material, energy and signal. This has severely limited its application to some generic systems, where such a flow of object is difficult to identify. Eder and Gosling [12] define such systems as *associative systems* (association of parts to form a whole unity). They pointed out that any attempt to define inputs and outputs for these systems is rather artificial. Umeda and Tomiyama [13] also argue that the input-output definition of function has trouble in representing a function that does not transform something, such as the function of a fixture, the function of a linear guide, and so on.

Correspondence and offprint requests to: Mr Y.-M. Deng, CAD/CAM Lab, School of Mechanical and Production Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798. Email: mymdeng@ntu.edu.sg

Our fundamental assumption is that mechanical function must be achieved through a certain action. Hence, we propose a design strategy based on a flow of *action*, and regard the *object* (material, energy or signal) as a specific attribute of *action*. In addition, we regard input-output flow transformation as a behavioural level of design abstraction. Hence, we use the input-output flow-of-action to represent *physical behaviour*, rather than *function*. These are discussed in detail in the following sections.

To support conceptual design, it is now a consensus that design information should include not only the physical structure of a design, but also its required function and implementing physical behaviour. For example, Sembugamoorthy and Chandrasekaran [14] present a functional design framework called Functional Representation (FR). FR is a model for describing the function of a device, its physical structure, and the causal process that culminates in the achievement of the function. Qian and Gero [15] present a model called the FBS (Function-Behaviour-Structure) path for design information and knowledge representation. Tomiyama et al. [16] present a Function-Behaviour-State (FBS) model for representing and dealing with function and function-related design knowledge, where *State* refers to the physical structure of a design at a particular physical state during its behavioural process.

We argue that all the above models lack an explicit representation of the product's working environment (although some models have considered exogenous behavioural variables). This aspect of design information includes not only the environment, but also the relationships (including the interactions) between the environment and the desired product. In the designing process, the designer must deal with this information at various levels. Explicitly representing and dealing with it can unify the synthesis reasoning strategies at these different levels. Faltings [17] has also stressed the importance of environment information. He stated that the definition of function should be made in the context of any restrictions imposed by the device's environment in which it operates.

2. Abstraction of Functional Design Information

Functional design information refers to the relevant information of a design that contributes to the development of functional design solutions. It is the functional-level knowledge of the product being designed. We have suggested a comprehensive design representation model, called the FEBS (Function-Environment-Behaviour-Structure) model [18], in which four aspects of functional design information are incorporated, i.e. the physical structure, the working environment, the required function, and the intended behaviour. In the following, we elaborate on these aspects of functional design information, and discuss how they are abstracted.

2.1. The Physical Structure

In the FEBS model, the physical structure of the product being designed consists of physical components that contribute to the required performance function, excluding those contributing to the other types of function, such as the assembly function, manufacturing function, market function and maintenance function [19].

The characteristics of a physical structure are defined by *attributes*. An attribute has a name, a value and a unit, e.g. weight of 50 *N*. Attributes may be static, such as weight, volume and temperature; or dynamic, such as velocity and acceleration. Attributes may form an attribute hierarchy from more general attributes to more specific attributes. An example is the type of material. A particular type of material may have further attributes, such as specific weight and thermal conductivity, that are relevant to a specific design.

2.2. The Working Environment

Any physical system (product) has a surrounding environment. The environment consists of a number of *environmental elements*. Some environmental elements might have no interactions with the product, or do not affect the product's intended behaviour, and are thus trivial to the design. The *working environment* includes only those environmental elements that contribute to the product's function. These environmental elements will have attributes that affect the behaviour of the product, such as the weight, temperature and surface properties.

Two kinds of relationship exist between the product and its environmental elements, i.e. a *geometric relation* and a *physical interaction*. A geometric relation characterises the relationships between the product and its environment in the geometric aspect, namely, the spatial and assembly relation. Physical interaction refers to the situation where the environmental elements (*source environmental elements*) provide input actions to the product, and the product

provides output actions to the environmental elements (*target environmental elements*). These relationships might also have certain attributes, such as the amount of force involved and the direction of an action.

There is a special kind of target environmental element, which we refer to as the *object-being-processed* (OBP). For many design problems, the required function is to change these OBPs from one state to another, or to a series of discrete physical states (physical change); or even to change the initial OBPs into different objects (chemical change). For example, the OBP in mould design is the material to be moulded. The OBPs in various machine tools are the workpieces to be machined. When designing such devices, the designer should first analyse the desirable behaviour of the OBPs, and then determine the required behaviour of the device so that it can be achieved. For example, when designing a domestic washing machine, the designer must first know the desired behaviour of the laundry and the water, and then devise the behaviour of the washing machine to achieve its required function.

2.3. The Required Function

Function plays a guiding role in the exploration of functional design information. The overall function characterises the general purpose or intention of the product being designed. This function may need to be decomposed to a set of sub-functions or a hierarchy (several levels) of sub-functions in the design process.

In the mechanical product design domain, we view the semantics of function as related to the level of the design hierarchy with which the function is associated. Generally, the overall required function and some of its sub-functions at the upper levels of the design hierarchy are expressed as a design intention. Conversely, the lower-level sub-functions need to be implemented by certain physical behaviour. These sub-functions are thus both a design intention and an abstraction of behaviour.

In this paper, we focus on exploring the functional design information that is necessary for developing the lower-level design hierarchy. We consider the development of an upper-level function hierarchy as a process for formalising a design specification in a way that simplifies the design task by providing a functionally simpler set of design sub-tasks. We call this process *initial function decomposition*. As part of the design specification, the designer should specify the upper-level design hierarchy by focussing on functionally independent design sub-tasks. This argument is in accord with the well-known Independence Design Axiom – maintaining independence of functional requirements [20]. For example, the required function of a domestic washing machine is 'to wash clothing'. This function might be initially decomposed by focusing on the set of functionally independent sub-tasks, and breaking them into sub-functions such as 'to add water', 'to wash', 'to drain', 'to rinse' and 'to spin'. The sub-tasks associated with these sub-functions might need to be performed in a cycling style (i.e. with the repetition of certain sub-tasks).

As an abstraction of physical behaviour, the lower-level mechanical functions, including the lowest-level sub-functions from the initial function decomposition, should be associated with an action or be expressed as an action. For example, 'to add water' is associated with an action of 'to move the inlet valve in a certain way so that the inlet gate is open'; 'to wash' with 'to rotate the washing drum in a certain pattern of motion'; 'to drain' with 'to move the outlet valve in a certain way so that the outlet gate is open'; 'to spin' with 'to rotate the washing drum at high speed', and so on.

2.4. The Intended Physical Behaviour

A product's function is achieved through a certain behaviour or behaviours. Regarding the role of behaviour in achieving a function, the following points should be noted:

- Only under the working environment can a behaviour produce its function. For instance, if we try to use a screwdriver to undo a screw with a badly burred slot, the behaviour of the screwdriver cannot fulfil its function (twist out screw), because the head of the screwdriver has nothing to act against. On the other hand, in an unintended environment, a product might achieve a certain unintended function. For example, apart from being used as a tool for enabling a person to drink water, a cup could be used for measuring (containing an approximately standard amount of liquid); or be used as a paperweight [21]. Unintended function is not considered in this paper.
- A physical structure has many properties and can demonstrate many behaviours beyond those intended by the designer. For example, when a bearing is supporting a shaft, its behaviour includes not only that of supporting the shaft, but also many others, such as dimensional distortion

because of the force acting from the shaft, generation of heat because of the friction between the bearing and the shaft, and so on. The physical behaviour that can produce the required function is called *intended physical behaviour* (e.g. supporting the shaft). Hence, intended behaviour is functional-level design information.

Figure 1 shows the four aspects of functional design information that are incorporated into the FEBS design model.

3. Flow-of-Action Behavioural Representation

Unlike functional analysis and modelling of existing physical systems, the functional design information discussed above is not readily available in the conceptual phase of mechanical product design. The problems to be answered are: How they can be explored, and thus identified? In what sequence shall they be identified? Under what environment or situation can all four aspects of design information be mutually related, so as to facilitate the exploration process? To answer these questions, we first propose a 'flow-of-action' strategy for behavioural representation, and discuss sub-behavioural-level functional design information.

3.1. Input-Output Flow-of-Action

When a product is in operation, physical interactions occur between the product and its working environment, as well as between the different components of the product. Hence, we suggest formalising these physical interactions to represent behaviour, i.e. to use an *input-output flow-of-action* behavioural representation. There are two kinds of input action: *driving input* (intended) and *harmful input* (unintended, which will potentially either reduce the efficiency of an intended behaviour or oppose it). Similarly, two kinds of output actions exist: *functional output* (intended) and *side-effect* (unintended, which is undesirable to the environment or performance of the product). For example, in producing light (function), a lamp might also produce heat. If the heat is not needed by the user or the designer, then 'producing heat' becomes a side-effect.

The following is a simple design example to show the difference between the commonly-used 'flow-of-object' and our proposed 'flow-of-action'.

Example 1 Considering the behaviour of a domestic washing machine, the input and output flow-of-object is:

- ***input***: *dirty laundry to be washed, clean water, detergent and electricity*;
- ***output***: *clean laundry, dirty water with detergent, etc.*

The input and output flow-of-action is:

- ***driving input***: *the start action by user or customised timer; the power supply, e.g. electricity supply; the support action from the floor where the washing machine is installed*;
- ***OBP***: *the laundry, the water and the detergent used*;
- ***functional output***: *the washing machine creates certain pattern of motion of the OBPs*;
- ***side-effect***: *the washing machine generates noise and produces dirty water.*

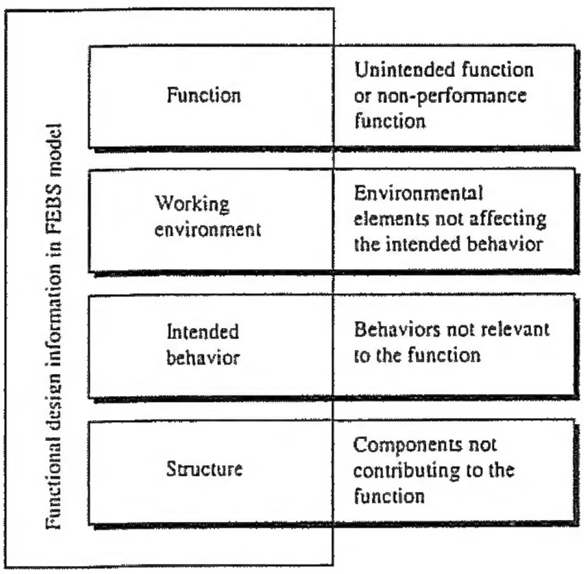

Fig. 1. Abstraction of functional design information.

3.2. Causal Behavioural Process

The intended behaviour performed by a product might consist of a set of sub-behaviours. Each sub-behaviour may be exhibited by a part or a whole of the product's physical structure. The sub-behaviours can be exhibited simultaneously (temporally parallel), or one after another (temporally sequential). For example, the behaviour of a paper punch is that the handle moves (exhibited by the handle), and at the same time, the two pins move (exhibited by the pins). These two sub-behaviours are exhibited temporally parallel, but logically in sequence. The causal sequence of these

sub-behaviours forms a causal process of the product's overall behaviour, which we call a Causal Behavioural Process (CBP).

The sub-behaviours of a causal behavioural process may be distinguished as being *output* and *internal*. We call those sub-behaviours that are to produce the product's output actions (i.e. its functional outputs or side-effects) *output sub-behaviours*; and the sub-behaviours that are to cause the output sub-behaviours to occur *internal sub-behaviours*. In the case of a washing machine, 'to rotate the washing drum' is an output sub-behaviour, while 'to produce rotary motion from a motor' and 'to produce the reduced rotary motion from a speed-reducer' are two internal sub-behaviours that are to 'drive' this output sub-behaviour.

Figure 2 shows an example of such a causal behavioural process. This is a directed graph, called a *CBP graph*. The B nodes in the graph represent sub-behaviours, called *sub-behavioural nodes*; while E nodes represent *environmental nodes*. There are two output sub-behaviours, B9 and B10. B1–B8 represent the internal sub-behaviours.

A causal behavioural process is an essential means for identifying functional design information, as it bridges the gap between required function and the physical structure of the product, and it also provides a means by which all four aspects of functional design information can be mutually related and integrated. In the next section, we elaborate on the sub-behaviour-level functional design information.

3.3. Relations, Attributes and Constraints

3.3.1. Assembly Relations

Spatial and assembly relations between components show how the components are assembled or connected to construct the required physical structure. In terms of the connection style and degree of freedom of the assembled components, we proposes thirteen categories of assembly relations for mechanical design (see Appendix A): *Contact, Connect, Fit, Fix, Insert, Join, Mesh, Near, Pivot, Rotate, Slide, Support* and *Weld*.

3.3.2. Attributes of Input/Output Action

The characteristics of driving inputs and functional outputs are defined by attributes in a manner similar to the attributes of physical structure and environmental elements. The three fundamental quantities (object flows) (i.e. material, energy and signal) can be simply represented by the attributes of input-output actions. For example, a compression spring (denoted as *C*) has a functional output action (denoted as *F*) to its target environmental element (denoted as *TE*): *F* = *press TE by C*. One of the attributes of this functional output (denoted as *FV1*) could be: *FV1* = *force of F*.

Because one input-output action can have a number of attributes, and these attributes can be used to represent the flow of energy, material and signal, the input-output flow-of-action approach thus enables the designer to consider the design interactions in a more natural and compact manner than the input-output flow-of-object approach. For example, if an operator pushes a handle, then this action can be represented as one single input action using the input-output flow-of-action approach, where the force from the operator and the angular movement of the handle caused by the push are represented as two attributes of the action. In this way, the designer can simply consider the push action from the operator when he or she is conceiv-

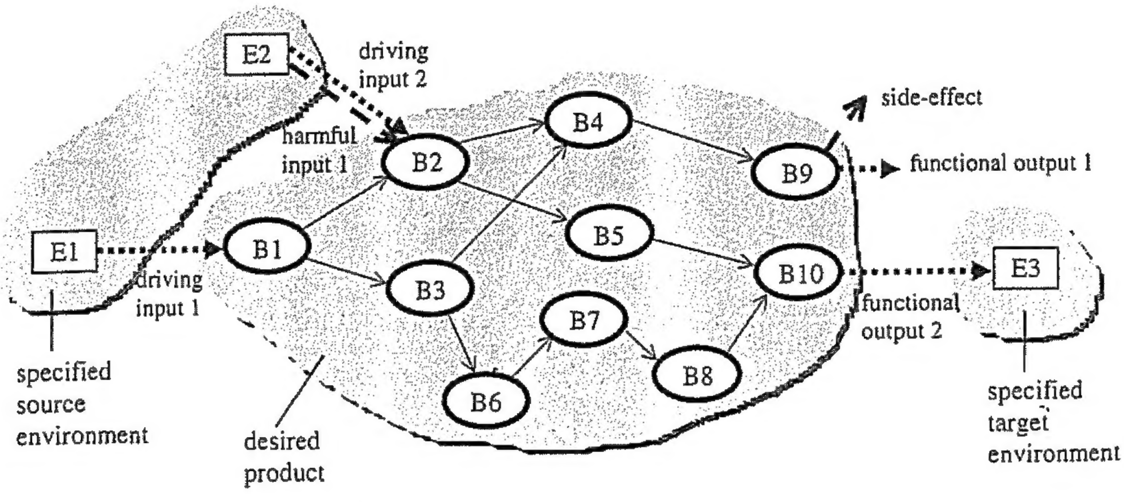

Fig. 2. A causal behavioural process graph.

ing design proposals. However, using the input-output flow-of-object approach, the force exerted by the operator and the angular motion of the handle must all be directly represented, hence the designer is burdened with these additional details during the design synthesis process.

3.3.3. *Attribute Expressions*

The attributes of functional outputs may be produced by physical structures or driving inputs. The relationship between their respective attributes can be expressed by physical laws. The expressions defining these physical laws are referred to as *attribute expressions*. As the attribute expressions characterise the significance of the functional outputs, they are thus also referred to as *functional output attribute expressions*. For the above example of a compression spring,

Physical structure:	C = *compression spring*
Attribute of C:	CV = *rate of C (spring rate)*
Driving input:	D = *compress C*
Attribute of D:	DV = *distance of D (compression distance)*
Functional output:	F = *produce force*
Attribute of F:	FV = *compression force of F*

The attribute expression of this functional output F is that $FV = CV * DV$, which is determined by Hooke's law.

3.3.4. *Phenomenon/Design Constraints and their Propagation*

For a physical structure to implement the required function, certain constraints may need to be satisfied. These constraints can act on the attributes of the physical structure, the working environment, as well as on the attributes of the required driving inputs and functional outputs. Furthermore, the attributes of harmful inputs and side-effects may also need to be constrained, to control their effect on the functional outputs. The list below gives some examples of such constraints:

- Constraints on the attributes of environmental elements: to initiate the execution of an intended behaviour, the temperature of the environment might have to exceed a certain value or change at a controlled rate.
- Constraints on the attributes of the physical structure: the radius of a fly wheel might have to exceed a certain value or the distance between two physical components might have to be within a certain range.
- Constraints on the attributes of driving inputs and functional outputs: the acting force might have to be greater than a certain value, the pressure might have to act in a certain direction, or the angular velocity might have to be within a certain range.

Each sub-behaviour corresponds to a certain physical phenomenon. Some of the constraints are required for the corresponding physical phenomenon to occur; these are referred to as *phenomenon constraints*. For example, if a sub-behaviour corresponds to a physical phenomenon called 'separation by friction', achieved by rotating a drum on top of a pack of paper, then the friction between the rotating drum and the paper pack is required in order for the paper pack to be separated into individual pieces. One of the required driving inputs is: to support and push the paper pack from below the paper pack. A constraint on this driving input attribute is: the pressure force should be maintained within a certain range. This constraint is used to guarantee that the phenomenon can occur.

Other constraints are used to determine whether the sub-behaviour can produce a functional output at a specified level or range of values, demanded by the specific design. These constraints are referred to as *design constraints*. For example, a design constraint on the paper separation sub-behaviour might be that the speed of separating the paper should be higher than a certain value.

Phenomenon and design constraints may be propagated along a behavioural process to give a set of *propagated constraints*. The propagated constraints are not new constraints, but equivalents of phenomenon constraints or design constraints at the different behavioural nodes or at the different parts of a behavioural node. Propagation is carried out in the inverse direction of the CBP digraph.

Within a sub-behaviour, phenomenon constraints only act on the attributes of physical structure or driving inputs, while design constraints only act on the attributes of functional outputs. In the causal behavioural process graph, the output sub-behaviours might have design constraints on their functional outputs, which can be directly derived from the design specification of an assigned problem. The constraint propagation process may be terminated at any internal connection between two sub-behavioural nodes where the preceding functional outputs can produce the required succeeding driving inputs, and also satisfy the related constraints; or the propagation may terminate at any sub-behavioural node where the propagated constraints are solely related

to the physical structure of the sub-behaviour, and are thus absorbed by the physical structure.

Table 1 summarises the aspects of functional design information of a sub-behaviour discussed above.

4. Exploring Functional Design Information by Flow-of-Action

4.1. Procedure for Exploring Functional Design Information

Based on the input-output flow-of-action strategy of behavioural representation, we suggest the following procedure for exploring functional design information:

1. *Formalisation of design specification.* The designer first formalises the design problem by an initial function decomposition based on the identification of functionally independent design sub-tasks. The process continues until all sub-tasks are sufficiently simple, and each lowest-level sub-function is associated with a certain action. The designer should also identify from the design problem specification the product's working environment and the relevant design constraints. The environmental information may be incomplete, as it can be explored in the later design process. For example, new environmental elements might need to be added (which might lead to modification of design specification or re-specification).
2. *Generation of causal behavioural process.* This enables the designer to explore what are the necessary sub-behaviours, their driving inputs, functional outputs, possible harmful inputs and side-effects, during which the physical structure for each sub-behaviour can be identified, including the assembly relations between the components of the sub-behaviour, and also between the components and the working environment. This is the central part of the exploration process. We have proposed a backward reasoning procedure for CBP generation [22], which is discussed with a design example in the next section.
3. *Construction of physical structure.* After all causal behavioural processes are generated, the physical structures for all sub-behaviours can be constructed, so that the physical structure of the whole product can be derived. The designer should then evaluate whether the design satisfies all of the design requirements. If not, then the whole procedure might need to be carried over again until feasible design can be derived.

4.2. A Design Example of Functional Design Information Exploration

Example 2 In designing a rivet setting device, the overall required function is 'to set rivet by a working head manually, and return the working head to its normal position automatically after each setting'. To facilitate our discussion, we first show one possible physical structure for the design in Fig. 3.

During initial function decomposition, it is assumed that two sub-functions associated with actions are identified – one is 'to exert certain force on the rivet by a working head, during the process the working head moves down a specified distance'; another is 'to move up the working head automati-

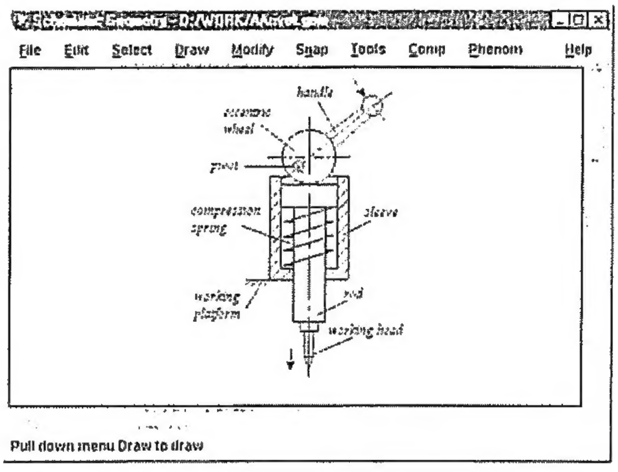

Fig. 3. A possible physical structure of a rivet setting device.

Table 1. Aspects of functional design information of a sub-behaviour

	Attributes	Attribute expressions	Phenomenon constraints	Design constraints	Propagated constraints
Physical structure	yes	no	yes	no	yes
Driving inputs	yes	no	yes	no	yes
Functional outputs	yes	yes	no	yes	yes

cally after each setting'. For the first sub-function, the associated design constraints are: 'the pressure force should be within a specified range' and 'the moving down distance should be equal to a specified value'. These design constraints are actually derived from certain design parameters relevant to the design assignment, for example, the material properties of the rivets, the required riveting force and stroke, and so on. The working environment is identified as consisting of the following environmental elements: the working platform, the operator of the device, and the rivet being set (an OBP).

After formalisation of the design specification, the designer needs to generate a causal behavioural process for the two sub-functions. In the following, we shall generate a CBP for the first sub-function, and use this as an example to illustrate the backward reasoning process for CBP generation.

Generate output sub-behaviours. The output sub-behaviours should have functional outputs which provide the actions associated with the sub-functions. For the current example, we denote the output sub-behaviour as B1. A physical component of *working head* is obviously required. Assume that the designer decides to use a *rod* (see Fig. 3), on which the working head is to be mounted, because it is predictable that during the operation process the working head should be constrained only to move vertically. Thus, the two physical components for B1 are: B1C1 = rod, B1C2 = working head. The functional output of B1 is just the required action for the first sub-function: B1F1 = exert a certain force on the rivet by a working head, during which the working head moves down a specified distance. The designer can easily determine the driving inputs: B1D1 = press the rod and working head; B1D2 = constrain the rod to move vertically. The attributes relating to this sub-behaviour and the constraints on the attributes should also be determined. Figure 4 illustrates the sub-behaviour.

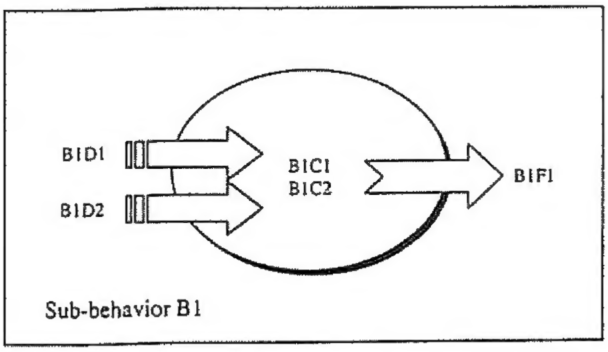

Fig. 4. Illustration of output sub-behaviour B1.

As can be seen, because of the use of flow-of-action, B1D1 (press the rod and working head) can abstract the design intention (press action) rather than the specific force and movement used in the flow-of-object approach. Similarly, B1D2 (constrain the rod to move vertically) can capture the design intention of a constraint action, which is difficult to represent by any flow-of-object.

Generate internal sub-behaviours. For each output sub-behaviour, determine whether its required driving inputs can be provided by the working environment, and if the associated constraints can be satisfied. If the answers to both questions are yes, then the output sub-behaviour does not require any internal sub-behaviour. If not, then one or more internal sub-behaviours are needed to 'drive' the output sub-behaviour. Reiterate this step for all output sub-behaviours, as well as all the internal sub-behaviours that have just been generated, i.e. generate other internal sub-behaviours to 'drive' these internal sub-behaviours. The process continues until all the initial driving inputs to the generated CBP can be provided by the working environment. During this process, the connections between the linked two sub-behaviours should also be established.

For the current example, the designer first resorts to the two source environmental elements: the human operator and the working platform. By using his or her hand, the operator can provide a press action on the device. However, the attribute of its functional output (i.e. the amount of force that can be produced by the operator) cannot satisfy the constraints on the driving inputs to B1 – the force should be within certain range, which is much greater than that a human operator can provide. Hence, internal sub-behaviours are required.

As each sub-behaviour corresponds to a certain physical phenomenon, the designer can either use his or her own expertise to figure out an appropriate physical phenomenon, or search and retrieve from a physical phenomena library stored *a priori* [22]. In any situation, as long as a known physical phenomenon is identified, the sub-behaviour can be generated. For the current example, one of the required internal sub-behaviours (indicated as B2) should be able to magnify the pressure force from the operator. Assume that the designer decides to use an *eccentric wheel* and *handle* to achieve this purpose (see Fig. 3): B2C1 = eccentric wheel; B2C2 = handle. Note that there might be many other solutions which are not expounded here for brevity. Another internal sub-behaviour (indicated as B3) should be able to constrain the movement of the

rod, assuming a *sleeve* is identified: B3C1 = sleeve. It can be seen that all the driving inputs to B2 and B3 can be provided by the two source environmental elements, hence no further internal sub-behaviour is required.

Examine harmful inputs and side-effects. The last step is to examine whether there is any significant side-effect and/or harmful input. If there is, then appropriate measures should be taken to nullify or alleviate it, such as by adding new internal sub-behaviours or by specifying another alternative design solution, such that this harmful input or side-effect won't occur. For the current example, no harmful input or side-effect is identified.

This completes the CBP generation for the first sub-function. Obviously, there might be alternative CBPs during the process. The designer should determine the best choice from these alternatives. With the selected causal behavioural process, the physical components for achieving the first sub-function would have been identified, e.g. B1C1, B1C2, B2C1, B2C2, B3C1. The assembly relations between these components would have been identified as well. After all CBPs for each sub-function are generated, the identified physical components can be assembled, during which all duplicating physical components from different CBPs should be excluded. Note that in Fig. 3, a *compression spring* was used. This physical component was determined in the CBP for the second sub-function.

Figure 5 shows a flowchart of the above-elaborated functional design information exploration procedure.

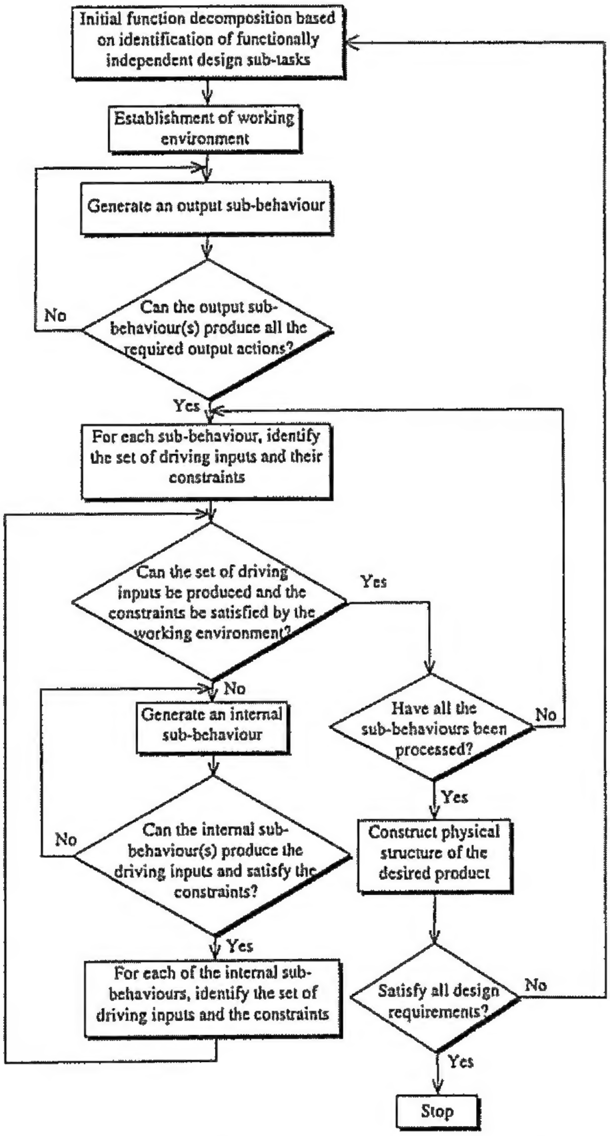

Fig. 5. Flowchart for functional design information exploration procedure.

5. Design Information Specification and Software Implementation

In this section we use a sub-module of a prototype functional modelling design environment [22] to illustrate the process of functional design information exploration and the specification of functional design information. The software prototype is also used to demonstrate the applicability of our input-output flow-of-action strategy.

In this prototype software, each sub-behavioural node of a CBP graph (indicated as B1, B2, etc.) is visualised as an oval attached with two small circles, where the oval represents the structural information of the sub-behaviour; its attached left circle represents the set of driving inputs; and the right circle represents the set of functional outputs. With similar denotations, the environmental nodes (indicated as E1, E2, etc.) are visualised as a rectangle, also with two small circles. Using this design environment, the designer first creates each node, specifies the functional design information to each node, and then connects the nodes to construct the causal behavioural process graph.

Figure 6 shows the CBP graph for the rivet-setting device example discussed above. E1 (human operator) and E2 (working platform) are two source environmental nodes. E3 (rivet) is a target environmental node. B1 (rod and working head) is the output sub-behavioural node. B2 (eccentric wheel and handle) and B3 (sleeve) are two internal sub-behavioural nodes. By pointing the mouse cursor onto the nodes, the designer can activate dialogue

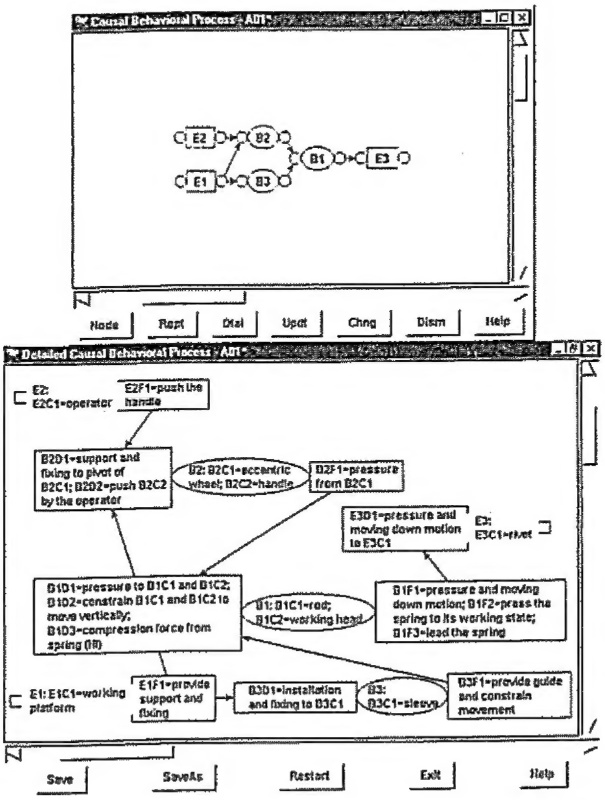

Fig. 6. The causal behavioural process for the rivet-setting device.

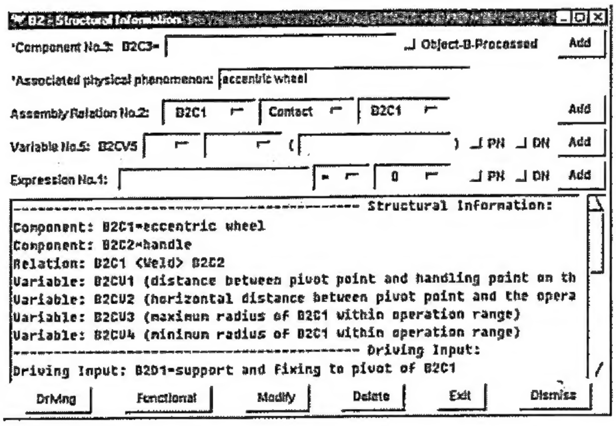

Fig. 7. Structural information for a sub-behaviour.

windows to specify the node's corresponding structural information, driving input information and functional output information, respectively. Taking sub-behavioural node B2 as an example, we illustrate in the following the relevant design information that needs to be specified.

5.1. Specifying Structural Information

Under the structural information dialogue window (Fig. 7), the following information should be identified and specified:

1. The components of the physical structure: B2C1 = eccentric wheel, B2C2 = handle. The identifying labels B2C1 and B2C2 are automatically generated by the software. Labels are used for generality. Designers can substitute actual names for the labels.
2. Associated physical phenomenon: if the designer has retrieved a physical phenomenon from a physical phenomena library to specify the sub-behaviour, then the physical phenomenon is associated with the sub-behaviour.
3. The assembly relations between the components: B2C1 <Weld> B2C2. The designer can select from the proposed thirteen commonly-used assembly types.
4. The attributes and constraints of the physical structure. A number of variables are used to represent the attributes, and expressions are used to represent the constraints on the attributes. Several commonly-used operators for the expressions are:

   = (equal to); > (greater than); < (less than); >= (greater than or equal to); <= (less than or equal to); != (not equal to).

   The right-hand-side value of an expression can be one of the following:

   0; specified value; specified range of value.

Some constraints might only relate to one variable. The software is designed to allow the designer to specify such constraints directly when specifying the relevant variables. For the current example, four variables are specified (see Fig. 7): B2CV1, B2CV2, B2CV3, B2CV4. There are propagated constraints from node B1, which are not discussed in this paper.

5.2. Specifying Driving Input Information

Under the driving input information dialogue window (Fig. 8), the following information should be identified and specified:

1. The set of driving inputs: B2D1 = support and fixing to pivot of B2C1 (eccentric wheel), B2D2 = push B2C2 (handle) by the operator. As some of the input actions might be harmful, the software is designed to allow the designer to specify this by clicking a check-button 'Harmful Input'. In the current example, no harmful input has been identified.

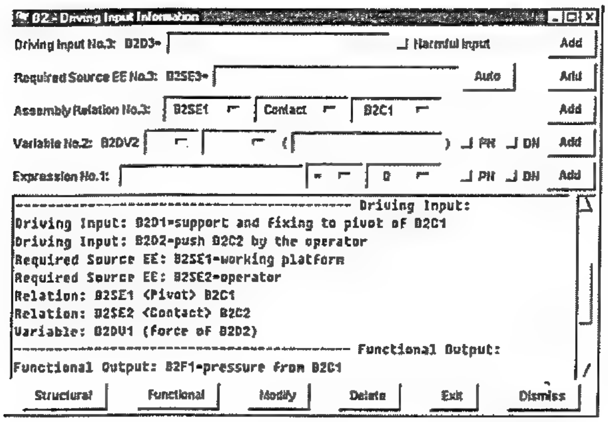

Fig. 8. Driving input information for a sub-behaviour.

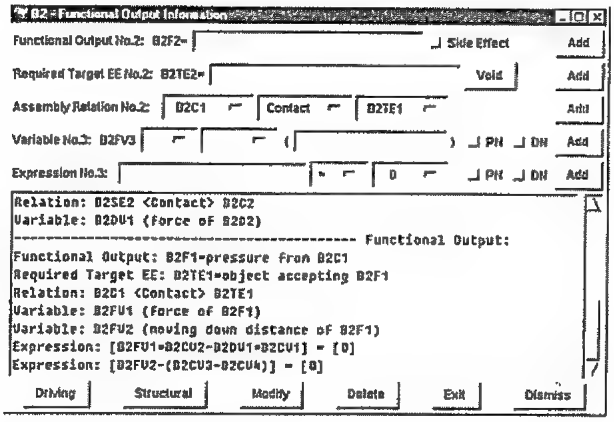

Fig. 9. Functional output information for a sub-behaviour.

2. The required source environmental elements: B2SE1 = working platform; B2SE2 = operator.
3. The assembly relations between the source environmental elements and the components of the physical structure. This enables the assembly relations between physical components of different sub-behaviours to be specified. For the current example, the assembly relations are: B2SE1 <Pivot> B2C1 (working platform <Pivot> eccentric wheel), and B2SE2 <Contact> B2C2 (operator <Contact> handle).
4. The attributes and constraints of the driving inputs. The only relevant attribute variable for the current example is: B2DV1 (force of B2D2, that is, the push force to the handle from the operator).

5.3. Specifying Functional Output Information

Under the functional output information dialogue (Fig. 9), the following information should be identified and specified:

1. The set of functional outputs: B2F1 = pressure from B2C1 (eccentric wheel). As some of the output actions might have side-effects, the software is designed to allow the designer to specify this by a check-button 'Side Effect'. For the current example, no side-effect is identified.
2. The required target environmental elements: B2TE1 = object accepting B2F1 (pressure from the eccentric wheel). B2TE1 is actually the rod and working head, which should be specified when specifying the connection information between node B2 and B1 (elaborated later on).
3. The assembly relations between the required target environmental elements and the physical components: B2C1 <Contact> B2TE1 (the eccentric wheel <Contact> the rod and working head).
4. The attributes, attribute expressions and constraints of the functional outputs. Two relevant variables are: B2FV1 (force of B2F1, i.e. the pressure force from the eccentric wheel to the rod and working head); B2FV2 (moving down distance of B2F1). The two expressions describing the attribute expressions of this functional output are:

- [B2FV1*B2CV2−B2DV1*B2CV1] = [0], and
- [B2FV2-(B2CV3−B2CV4)] = [0].

The former attribute expression shows that the force provided by the operator is transmitted to the force acting on the rod and working head, which is magnified by B2FV1/B2DV1 = B2CV1/B2CV2. The latter attribute expression signifies the moving down distance from the eccentric wheel. There are propagated constraints from node B1 (not to be discussed).

5.4. Internal Connection between Sub-Behavioural Nodes

The internal connection between two sub-behavioural nodes signifies the causality between them, i.e. the functional outputs from the preceding node are actually used as the driving inputs to the succeeding node. Similar connections occur between environmental nodes and sub-behavioural nodes, if causal relationships exist. Because one node may connect with more than one preceding node, and also with more than one succeeding node, the designer should specify the functional outputs from a preceding node that are actually used as the driving inputs of its succeeding node. The designer

needs to graphically connect the two causally related nodes, and also specify any other information relevant to the connection.

The connections for the above example can be seen in Fig. 6. We shall take the connection between node B2 and B1 as an example to explain how the connection information is specified (refer to Fig. 10).

Additional information not captured by the screen shot in Fig. 10 is listed below:

Sub-Behaviour Node B1 requires driving inputs
B1C1=rod
B1C2=working head
B1D1=pressure to B1C1 and B1C2
B1D2=constrain B1C1 and B1C2 to move vertically
***Harmful Input: B1D3 = compression force from spring[1]
B1F1=pressure and moving down motion
B1F2=press the spring to its working state
B1F3=lead the spring
B1SE1=object providing B1D1
B1SE2=object providing B1D2
B1C1 <Fix> B1C2
B1SE1 <Contact> B1C1
B1SE2 <Slide> B1C1

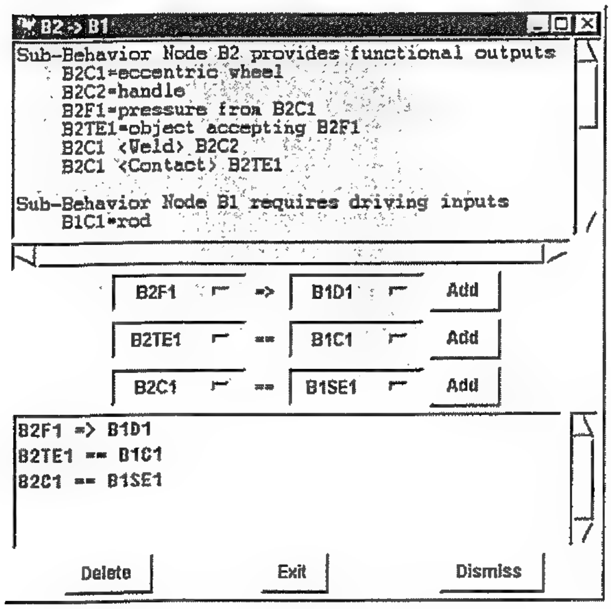

Fig. 10. Specifying connection information between node B2 and B1.

[1] The harmful input B1D3 is caused by the compression spring, which resists the movement of the rod and working head during the rivet setting process. The compression spring is used for the second sub-function, i.e. automatically returning the rod and working head after each setting process.

The specification window is designed to list the relevant information of both the preceding node and the succeeding node. With reference to this information, the designer can match the following pairs:

1. The functional outputs from the preceding node and the driving inputs to the succeeding node:

   B2F1 => B1D1 (pressure from eccentric wheel => pressure to rod and working wheel).

2. The required target environmental elements of the preceding node and the physical components of the succeeding node. This allows the designer to examine whether the preceding node's required target environmental elements can be provided or not:

   B2TE1 == B1C1 (object accepting pressure from eccentric wheel == rod).

3. The physical components of the preceding node and the required source environmental elements of the succeeding node. This allows the designer to determine whether the succeeding node's required source environmental elements can be provided or not:

   B2C1 == B1SE1 (eccentric wheel = object providing pressure to rod and working head).

4. The attributes (variables) of the functional outputs of the preceding node and the attributes (variables) of the driving inputs of the succeeding node. This matching is done automatically, because the matching between the functional outputs of the preceding node and the driving inputs of the succeeding node has already been specified, and each variable has an established link with the corresponding functional output or driving input. Figure 11 demonstrates an example of such matching. This matching is important, as it

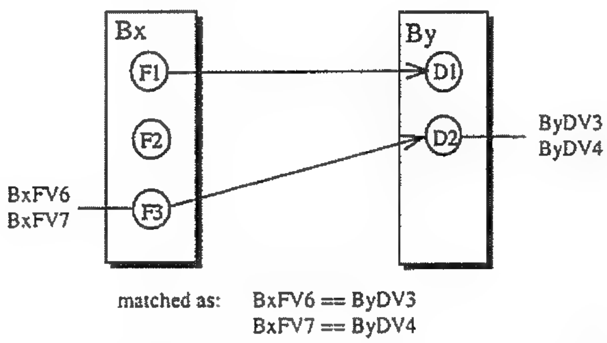

Fig. 11. Automatic matching between attributes of the matched pairs of functional outputs and driving inputs.

allows the design constraints on the driving inputs of a succeeding sub-behaviour to be either propagated through or terminated at the internal connection with its preceding sub-behaviour. For the current example, the following match is identified by the design environment:

B2FV1 (force of pressure from eccentric wheel) = B1DV1 (force of pressure to rod and working head);

B2FV2 (moving down distance by eccentric wheel) = B1DV2 (moving down distance of rod and working wheel).

The above specification is applicable when no harmful input or side-effect information is involved in either of the two connected sub-behaviours. For connection between two nodes which have harmful inputs or side-effects, the following rules apply:

- *Rule 1*: the functional output of a preceding node can match with the side-effect of a succeeding node, which signifies that the functional output of the preceding node is used to prevent the occurrence of the succeeding node's side-effect.
- *Rule 2*: the functional output of a preceding node can match with the harmful input of a succeeding node, where the harmful input might come from another sub-behavioural node or another environmental node. This signifies that the harmful input is nullified by this preceding node's functional output.
- *Rule 3*: the side-effect of a preceding node can match with the harmful input of a succeeding node, which means that the preceding node's side-effect is actually instantiated as the harmful input to the succeeding node. If this is the case, the designer should ensure that the relevant attributes of the harmful input are within the constraint range, i.e. the harmful input will not influence the positive functioning of the corresponding sub-behaviour.

For facilitating the specification of connection information under the above situations, the software is designed to list both the driving inputs and functional outputs of the succeeding sub-behavioural node, as can be seen in the list that refers to Fig. 10.

The software is also designed to report the connection information, so that the designer can examine whether there is any driving input to a sub-behaviour that has not been satisfied; or any functional output that has not been used by other sub-behavioural nodes; or any required source or target environmental element that cannot be provided by the product or its working environment. This instant feedback will provide the designer with a useful guidance as he or she is exploring the further functional design information. The report also summarises what are the physical components that have been identified, and how they are assembled or connected, together with the product's working environment information. A report for the current design example is listed in Appendix B, where the first part reports the known summary information, and the second part reports the unknown information, notifying the designer to continue his or her exploration.

During and after this design information exploration process, the designer can employ the software to check the four aspects of functional design information that have already been identified. A report of function information for the current example is listed in Appendix C.

6. Conclusions

Our discussion in this paper is focused on: What is the essential functional design information (abstraction), and how this information can be explored and identified (exploration)? Four aspects of functional design information are discussed: the required function; the working environment; the causal behavioural process; and the physical structure. Physical behaviour is represented by an input-output flow-of-action, and this is used in the exploration of functional design information of a product being designed.

Causal behavioural process generation is recognised as an essential means by which all aspects of functional design information can be explored in an integrated manner. A design example is used to illustrate how these various aspects of design information can be explored and determined. This is assisted by a software prototype developed by us. Based on these discussions, we highlight a number of conclusions:

1. It is important to distinguish between functional and non-functional design information, as it allows a designer to abstract only the design information that is necessary and essential for the development of functional design solutions. Functional-level design information offers a designer both a more focused design effort as well as a wider range of design selection.
2. A mechanical function is regarded as relating to the level of the design hierarchy with which the function is associated. The upper-level functions are generally a design intention. Function decomposition at the upper levels can be carried out by exploring the functionally independent design sub-

tasks. The lower-level functions are both a design intention and an abstraction of physical behaviour. Developing a lower-level design hierarchy requires the designer to explore the causal relations between the design sub-tasks (achieved by sub-behaviours), for which we suggest generating the product's causal behavioural process.

3. The input-output flow-of-action strategy allows designers to synthesize design problems where there is no distinct flow-of-object, hence it is more generic. It can not only deal with a wider range of physical devices, but also distinguish between the intended and unintended (harmful) input, as well as between the intended and unintended output (side-effect). The causal behavioural process was originally proposed in the Functional Representation (FR) model [14]. However, the FR model abstracts physical behaviour as state transitions, which narrows its applicability. Not only is our approach more generic, it facilitates the exploration of design information because of the embodiment of causality by the functional output and driving input pairs.
4. To facilitate the exploration of functional design information, we also abstract functional design information at the sub-behavioural-level, including the assembly relations, attributes, attribute expressions and constraints (both phenomenon constraints and design constraints) relevant to the physical components, the working environment, the driving inputs, and the functional outputs. This detailed information has been shown as useful in the exploration of functional design solutions.

Although the proposed input-output flow-of-action strategy has been demonstrated to have a strong reasoning capability during the CBP generation process, synthetic reasoning design needs to be conducted manually. Hence, work could be done to provide automatic reasoning for the design synthesis process. This in turn requires a strategy for automatically searching and retrieving the background functional design knowledge. We have developed a functional modelling design environment based on the work described in this paper [22]. Currently, we are improving the design environment by upgrading it into a knowledge-based functional design system. A three-level functional design knowledge architecture has been developed, and an automatic reasoning strategy for the system is under research. We are also collaborating with local companies in Singapore to put our work into practical use in the field of consumer-product designs.

References

1. Tor, S. B., Britton, G. A., Chandrashekar, M., Ng. K. W. (1998) Functional design. In: Usher, J. M. *et al.* (Editors). Integrated Product and Process Development: Methods, Tools and Technologies. Wiley, New York, 29–58
2. Xue, D., Takeda, H., Kiriyama, T. (1992) An intelligent integrated interactive CAD – A preliminary report. In: Brown, D. C., Waldron, M., Yoshikawa, H. (Editors), Intelligent Computer Aided Design. Elsevier, Amsterdom, 163–187
3. Pahl, G., Beitz, W. (1988) Engineering Design: A Systematic Approach. Springer-Verlag, London
4. Hubka, V., Eder, W. E. (1988) Theory of Technical Systems. Springe-Verlag, London
5. Schmekel, H. (1989) Functional models and design solutions. Annals of the CIRP, 38(1), 129–132
6. Hundal, M. S. (1990) A systematic method for developing function structures, solutions and concept variants. Mechanism and Machine Theory, 25(3), 243–256
7. Welch, R. V., Dixon, J. R. (1992) Representing function, behavior and structure during conceptual design. Design Theory and Methodology, ASME, 42, 11–18
8. Kuttig, D. (1993) Potential and limits of functional modelling in the CAD process. Research in Engineering Design, 5, 40–48
9. Chakrabarti, A., Bligh, T. P. (1994) An approach to functional synthesis of solutions in mechanical conceptual design, Part I: Introduction and knowledge representation. Research in Engineering Design, 6, 127–141
10. Kota, S., Lee, C. L. (1993) General framework for configuration design: Part I – Methodology. Journal of Engineering Design, 4(4), 277–294
11. Ullman, D. G. (1993) A new view on function modeling. Proceedings of ICED'93, The Hague, 21–28
12. Eder, W. E., Gosling, W. (1965) Mechanical System Design. Pergamon, Oxford
13. Umeda, Y., Tomiyama, T. (1997) Functional reasoning in design. IEEE Expert, 12(2), 42–48
14. Sembugamoorthy, V., Chandrasekaran, B. (1986) Functional representation of devices and compilation of diagnostic problem-solving systems. In: Kolodner, J., Riesebeck, C. (Editors), Experience, Memory, and Reasoning. Lawrence Erlbaum, Hillsdale, NJ, 47–73
15. Qian, L., Gero, J. S. (1996) Function-behavior-structure paths and their role in analogy-based design. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 10, 289–312
16. Tomiyama, T., Umeda, Y., Yoshikawa, H. (1993) A CAD for functional design. Annals of the CIRP, 42(1), 143–146
17. Faltings, B. (1991) Qualitative models in conceptual design. Proceedings of the First Conference on Artificial Intelligence in Design, 645–663
18. Tor, S. B., Deng Y.-M., Britton, G. A. (1999) A comprehensive representation model for functional design of mechanical products. Proceedings of the 12th International Conference on Engineering Design, Munich, Germany, 1929–1932
19. Deng Y.-M., Britton, G. A., Tor, S. B. (1998) A design perspective of mechanical function and its object-oriented representation scheme. Engineering with Computers, 14, 309–320

20. Suh, N. P. (1990) The Principles of Design. Oxford University Press, New York
21. Rosenman. M. A.. Gero, J. S. (1994) The what, the how, and the why in design. Applied Artificial Intelligence. 8(2), 199–218
22. Deng Y.-M., Tor, S. B., Britton, G. A. (1999) A computerized design environment for functional modeling of mechanical products. Proceedings of the 5th ACM Symposium on Solid Modeling and Applications, Ann Arbor, MI, 1–12

Appendix A. Categories of Assembly Relations

Assembly	Explanation	Examples
1. Contact	two or more components are in physical contact and the contact status must be maintained so that the pressure force between the components is within certain range	a cam and cam-follower, an eccentric wheel and its activated physical component
2. Connect	two components are connected but not in direct contact, i.e. the components are separated but connected via one or more intermediate components	an electricity supply and electric motor, two wheels connected by a belt
3. Fit	two components are directly connected with each other without using any intermediate component, and they are also in contact with each other without using pressure force	the snap-fitted components, two or more pipes fitted together (pipe-fitting)
4. Fix	one component is fixed to another so that there is no mutual movement between them. The two components, however, are detachable	two components that are connected with screws, or a key, or a pin
5. Insert	one component is inserted into another component so that either the former can only move in the confine of the latter; or the latter can only move without escaping from the former	a compression spring and its guiding rod
6. Join	two components are joined together so that both of them can rotate around the joint and the joint can move as well	two links joined at their ends
7. Mesh	this specifically refers to the relationship between two gears	
8. Near	two components are not in direct contact or connected, yet there is interaction between them caused by a field (e.g. magnetic field). Such a relationship often requires the distance between the two components to be maintained at a certain value or within a certain range	the rotor and stator of a motor
9. Pivot	one component is fixed on another by a pivot point. so that the first component can rotate around the pivot. while the second is usually static	an eccentric wheel and the working platform upon which the eccentric wheel is mounted, a lever and its pivot holder
10. Rotate	one component is inserted or partly inserted in another component, so that the mutual movement between them is restricted to rotary motion	a shaft and bearing
11. Slide	one component is inserted or partly inserted in another component. so that the mutual movement between them is restricted to sliding motion	a slider and guide
12. Support	one component is supported by another so that the former must be on top of the latter. It is a special case of both the 'Contact' relation and the 'Fix' relation	the paper on the table and the table that supports the paper
13. Weld	two components are welded or soldered together so that they act as one single component. Different from 'Fix' relation, the two components are non-detachable	

Appendix B. Report of Summary Information of Design Example 2

== Known Summary Information of the CBP of Alternative Design A01 ==
All physical components:

No.1: B1C1=rod
No.2: B1C2=working head
No.3: B2C1=eccentric wheel
No.4: B2C2=handle
No.5: B3C1=sleeve

All environmental elements:

No.1: E1C1=working platform
No.2: E2C1=operator
No.3: E3C1=rivet

Assembly relations between physical components:

No.1: B1C1 <Fix> B1C2
No.2: B2C1 <Contact> B1C1
No.3: B3C1 <Slide> B1C1
No.4: B2C1 <Weld> B2C2

Assembly relations between physical components and environmental elements:

No.1: E1C1 <Pivot> B2C1
No.2: E1C1 <Fix> B3C1
No.3: E2C1 <Contact> B2C2
No.4: B1C2 <Contact> E3C1

Exterior driving inputs from source environmental elements:

No.1: E1F1=provide support and fixing => B2D1=support and fixing to pivot of B2C1
No.2: E1F1=provide support and fixing => B3D1=installation and fixing to B3C1
No.3: E2F1=push the handle => B2D2=push B2C2 by the operator

Required driving inputs for target environmental elements:

No.1: B1F1=pressure from B1C2 => E3D1=pressure to E3C1
No.2: B1F2=move down a specified distance => E3D2=move down E3C1 certain distance

Overall functional outputs:

No.1: B1F1=pressure from B1C1
No.2: B1F2=move down a specified distance
No.3: B1F3=press the compression spring

Overall functional outputs not used for target environmental elements:

No.1: B1F3=press the compression spring

== Unknown Information of the CBP of Alternative Design A01 ==
Unused functional outputs:

None

Unsatisfied driving inputs:

None

Unspecified source environmental elements:

None

Unspecified target environmental elements:

No.1: B1TE2=compression spring

Appendix C. Report of Function Information of Design Example 2

== Function Information ==

The overall required function:
F111: set rivets manually
return the working head automatically after each setting

Sub-functions from selected FDM alternative design:
F211: exert certain force on the rivet
move down the working head a specified distance
F212: move back the working head a specified distance

— From CBP alternative design A01 —

Sub-function F211 decomposed into:
B1: pressure from the working head
move down a specified distance
press the compression spring
B2: pressure from the eccentric wheel
move down a certain distance
B3: provide guide and constrain movement

— From CBP alternative design A02 —

Sub-function F212 decomposed into:
B1: move back the working head a specified distance
provide lifting force
B2: provide lifting force
B3: rotate the eccentric wheel and the handle to return them back

Glossary

Functional design: refers to a general area of research that focuses on the early phases of the

design process and designing by function. The aim of functional design is to provide computer tools to link design functions with structural (physical) embodiments used to realise the functions [1]. Depending on particular situations, functional design also refers to an early-stage design process or certain design activities, which aims at finding functional design solutions.

Function: the abstract information characterising the intention of the designer for a product. This intention describes the intended task, activity or work which the product will be able to perform after it has been designed and manufactured.

Structure: physical structure refers to a physical system, a sub-assembly, a component or a number of components, a feature, or a geometric entity and physical relationships, depending on the specific design context. For early-stage mechanical product design, it often refers to a set of components that are functionally integrated, which may or may not be a sub-assembly or sub-system of the design.

Behaviour: physical behaviour refers to the operational process of a product under its working environment. In this paper, it is specifically referred to as the physical interactions between the components of the product, as well as between the product and its working environment.

Working environment: includes those environmental elements of a product that contribute to the product's function or intended behaviour.

Object-being-processed (OBP): a special type of target environmental element. For many design problems, the required function is to change the OBPs from one state to another, or to a series of discrete physical states (physical change); or to change the initial OBPs into some different objects (chemical change).

Input-output flow-of-object: refers to a flow of material, energy or signal that is commonly used in the analysis and synthesis of physical systems.

Input-output flow-of-action: during a product's behavioural process, physical interactions occur between the product and its working environment, as well as between the components of the product. Input-output flow-of-action refers to the formalised physical interactions, including driving inputs, functional outputs, as well as possible harmful input and side-effect.

Driving input: an intended input action required by a product or a sub-behaviour of the product in order that a required function or sub-function can be achieved.

Functional output: an intended output action produced by a product or a sub-behaviour of the product.

Harmful input: an unintended input action to the product or its sub-behaviour, which either potentially reduces the efficiency of the intended behaviour or oppose it.

Side-effect: an incidentally produced output action (unintended), and is undesirable to the environment or to the performance of the product or its sub-behaviour.

Causal behavioural process: refers to a network of sub-behaviours that defines the causal sequence of the product's sub-behaviours in achieving a required function or functions.

Output and internal sub-behaviour: output sub-behaviours refer to the sub-behaviours that produce the product's output actions (i.e. its functional outputs or side-effects). Internal sub-behaviours refer to the sub-behaviours that produce the required driving inputs to the output sub-behaviours.

Assembly relation: spatial and assembly relation (simply called *assembly relation*) refers to the relationship between two or more components that define how the components are assembled or connected to construct a physical structure.

Attribute: defines the characteristics of an object (including a physical structure, an environmental element, or a driving input/functional output). An attribute has a name, a value and a unit, e.g. weight of 50 N.

Attribute expression: the relationship between the attributes of a physical structure, its driving input and functional output can be expressed by physical laws. The expressions defining these physical laws are referred to as attribute expressions.

Phenomenon constraint: each sub-behaviour of a design corresponds to a physical phenomenon. Phenomenon constraints refer to the constraints on the attributes of the sub-behaviour that are required for the corresponding physical phenomenon to occur.

Design constraint: refer to the constraints on the attributes of a sub-behaviour that are required for the sub-behaviour to produce a functional output at the desired level or range of value.

Propagated constraint: phenomenon and design constraints may be propagated along a causal behavioural process to give a set of propagated constraints. They are equivalents of phenomenon constraints or design constraints at the different behavioural nodes of a design or at the different parts of a behavioural node.